Frequent Pattern Mining Algorithm Based On Single Scanning Of Whole Transactional Dataset

Parag M. Moteria
PhD Scholar,
School of Computer Science,
R K University, Rajkot.

Dr. Y R Ghodasara
Associate Professor

Abstract

Frequent patterns are frequent data set in transactional data set, play an essential role in mining associations, correlations and many other interesting relationships among data that leads knowledge discovery and helps in many business decision making processes [1]. Data mining is a very basic operational technique in knowledge discovery and decision making processes. Frequent pattern mining techniques have become necessary for massive amount datasets in data mining approach. This paper discuss algorithm for efficient and scalable frequent itemsets mining on Boolean types of single dimensional and single level data mining in transactional dataset through only one time scanning whole transactional dataset.

1. INTRODUCTION

Data mining is the process of finding interesting trends or patterns in large datasets to steer decision about future activities. Knowledge discovery in databases and data mining helps to extract useful information from raw data. Frequent itemsets play an essential role in many data mining tasks that try to find interesting patterns from databases or transactional dataset, such as association rules, correlations, sequences, episodes, classifiers, clusters. Frequent pattern mining is one of the most important and well researched techniques of data mining. Association rules can be useful for decisions concerning product pricing, promotions, store layout and many others [2]. Thus, frequent pattern mining has become an important data mining task and a focused theme in data mining research [3]. Our frequent pattern mining algorithm is data mining where computations are done on transactional dataset scan whole dataset only once. This paper describes frequent pattern mining algorithm that scan whole transactional dataset only once.

2. PROBLEM STATEMENT

Multiple scanning processes of whole transactional datasets to determine k numbers of itemsets from each transaction is major problem. Scanning process takes major time to determine frequent pattern itemsets. After determination of frequent pattern itemsets, another problem is to generate association rules and confidence rule, efficiently. Consider, TID is an identifier in transactional datasets say D. Association rule notation for two independent itemset is X => Y.

$$\text{Support } (X \Rightarrow Y) = \sup(XY) / |D|$$
$$\text{Confidence } (X \Rightarrow Y) = \sup(XY) / \sup(X)$$

A frequent itemset is an itemset whose occurrence is frequently in transactional datasets that is above minimum user defined support threshold.

3. THE APRIORI ALGORITHM [1]

Consider following transactional dataset in lexicographic order [1] and user minimum support count is 2:

Table 1

TID	List of ITEM IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

After applying the apriori algorithms, following tables shown results after each consecutive step to mine frequent itemsets from Table1.

Table 2	
C1	
Itemset	Support Count
I1	6
I2	7
I3	6
I4	2
I5	2

Table 3	
L1	
Itemset	Support Count
I1	6
I2	7
I3	6
I4	2
I5	2

Table 4	
C2	
Itemset	Support Count
I1,I2	4
I1,I3	4
I1,I4	1
I1,I5	2
I2,I3	4
I2,I4	2
I2,I5	2
I3,I4	0
I3,I5	1
I4,I5	0

Table 5	
L2	
Itemset	Support Count
I1,I2	4
I1,I3	4
I1,I5	2
I2,I3	4
I2,I4	2
I2,I5	2

Table 6	
C3	
Itemset	Support Count
I1,I2,I3	2
I1,I2,I5	2

Table 7	
L3	
Itemset	Support Count
I1,I2,I3	2
I1,I2,I5	2

We get frequent itemsets, {I1,I2,I3} and {I1,I2,I5} which satisfy user minimum support threshold is equal to 2 using the apriori algorithm after multiple scanning of transactional datasets.

4. PROPOSED FREQUENT PATTERN MINING ALGORITHM BASED ON SINGLE SCANNING OF WHOLE TRANSACTIONAL DATASET (FPMA-SS)

The apriori algorithm, we have to count the support of itemsets many times during mining process. Since counting the occurrences of itemsets is a time-consuming process. Hence, the present paper proposes algorithm for mining frequent patterns that scans whole datasets only once. In case of Apriori algorithm when we count the support of candidate set of length k, we also check its occurrence in transaction whose length may be greater than, less than or equal to the k[2]. But in the proposed algorithm support count of each itemset using binary matrix generated from transactional datasets. This binary matrix helps to calculate support count of each itemsets with their appropriate cardinality. Here, term cardinality represents numbers of items in transactions.

Our proposed algorithm achieves same result as per above discussion based on the apriori algorithm only single time scanning of whole transactional dataset through following steps: (Consider Table1)

Step 1:
Applying this step, we generate binary matrix of whole transactional datasets. Binary matrix is shown in Table 8.

Table 8						
TID	I1	I2	I3	I4	I5	Cardinality
T100	1	1	0	0	1	3
T200	0	1	0	1	0	2
T300	0	1	1	0	0	2
T400	1	1	0	1	0	3
T500	1	0	1	0	0	2
T600	0	1	1	0	0	2
T700	1	0	1	0	0	2
T800	1	1	1	0	1	4
T900	1	1	1	0	0	3
Total	6	7	6	2	2	

After getting Table8, we get total numbers (Support Count) of each item in last row that is similar result as per Table2.

Step 2:
This step, calculate cardinality wise total numbers of each item. Each column heading shows cardinality.

Table 9

	1	2	3	4	5	
I1	0	2	3	1	0	6
I2	0	3	3	1	0	7
I3	0	4	1	1	0	6
I4	0	1	1	0	0	2
I5	0	0	1	1	0	2

Step 3:
Applying this step, we get cardinality wise actual support count of each item. that is similar result as shown in Table4 (Cardinality=2) and Table6 (Cardinality=3).

Table 10

	1	2	3	4	5	
I1	0	4	4	1	0	6
I2	0	5	4	1	0	7
I3	0	4	2	1	0	6
I4	0	1	1	0	0	2
I5	0	0	2	1	0	2

Step 4:
Verify user minimum support count threshold for each item in Table10 from highest cardinality column. Those items do not satisfy minimum support count threshold, remove from Table10, we get following resultant table.

Table 11

	1	2	3	
I1	0	4	4	6
I2	0	5	4	7
I3	0	4	2	6
I5	0	0	2	2

Table11 shows similar result as describe in Table4 (Cardinality = 2) and Table6 (Cardinality = 3).

Our proposed algorithm yields same frequent items say I1, I2, I3 and I5, those are result of apriori algorithm.

The apriori algorithm:
Table4 (Cardinality = 2),
Support count of {I1,I2} = 4

Proposed algorithm:
Table11 (Cardinality = 2),
Support count of {I1,I2} = Min{4,5} = 4

The apriori algorithm:
Table6 (Cardinality = 3),
Support count of {I1, I2, I3} = 2

Proposed algorithm:
Table11 (Cardinality = 3),
Support count of {I1, I2, I3} = Min{4,4,2} = 4

Only single scan of transactional dataset is required using our proposed algorithm to yield support count of each item in transactional datasets.

5. USE OF RESULTANT TABLE 11 TO CALCULATE CONFIDENCE COUNT

We get {I1, I2, I3, I5} as frequent itmeset in result. Power set of these resultant frequent itemset consists sixteen different unique combinations to calculate confidence count.

For example,

Conf(I1 => I2) = (sup(I1,I2)) / sup(I1)
= Min {sup (I1), sup (I2)} / sup (I1)
= Min {4, 5} / 6
= 4 / 6

In above,

sup (I1, I2) , there are two items in support count. Therefore, we refer column with cardinality two from Table11. When we have single item in support count, it refers total count of specific item from the Table11.

6. RESULT

We represent processing time with respect to different transactional datasets with different maximum numbers of cardinality.
To explore the performance of proposed algorithm, synthetic dataset is used and all the experiments are performed on Intel(R) Core (TM)2 Duo CPU 2.93 GHz PC machine with 2 GB RAM, running Microsoft Windows XP service pack 3.

Table 12

Minimum Support Count = 2

Nos. of Record	Maximum Nos. of Cardinality	Processing Time (In Milliseconds)
1620	4	203
1868	18	237
1967	5	202
1845	4	206
98	21	160

5. Conclusion and Future Work

Our proposed algorithm scan of whole transactional dataset only once and yields efficient and scalable frequent itemset mining to enhance strength to discover knowledge. It may save processing time and resources. Our future work is continuously improving efficiency of our proposed algorithm and maintains accuracy to count confidence.

References

[1] Jiawei Han and Micheline Kamber, Data Mining Concepts and Techniques - Third Edition, ELSEVIER Morgan Kaufman Publisher, July 6, 2011

[2] D. N. Goswami, Anshu Chaturvedi, C. S. Raghuvanshi, "Frequent Pattern Mining Using Record Filter Approach",International Journal of Computer Science, Vol. 7, Issue 4, No 7, July 2010, pp 38-43

[3] Jiawei Han, Hong Cheng, Dong Xin, Xifeng Yan, "Frequent pattern mining: current status and future directions", Springer Science+Business Media, LLC 2007, pp 55-86

[4] Anjan K Koundinya,Srinath N K,K A K Sharma, Kiran Kumar, Madhu M N and Kiran U Shanbag, "Map/Reduce Design And Implementation Of Apriorialgorithm For Handling Voluminous Data-Sets", ACIJ, Vol.3, No.6, November 2012, pp 29-39

[5] Lamine M. Aouad, Nhien-An Le-Khac and Tahar M. Kechadi, "Distributed Frequent Itemsets Mining in Heterogeneous Platforms", Journal of Engineering, Computing and Architecture, Vol. 1, Issue 2, 2007

[6] Bagrudeen Bazeer Ahamed and Shanmugasundaram Hariharan, "A Survey On Distributed Data Mining Process Via Grid", International Journal of Database Theory and Application, Vol. 4, No. 3, September 2011, pp 77-90

[7] Goswami D.N., Chaturvedi Anshu., Raghuvanshi C.S.,"An Algorithm for Frequent Pattern Mining Based On Apriori", IJCSE, Vol. 02, No. 04, 2010, pp 942-947

[8] Sunil Joshi, R S Jadon and R C Jain, "A Frame Work for Frequent Pattern Mining Using Dynamic Function", IJCSI, Vol. 8, Issue 3, No. 1, May 2011, pp 141-147

[9] Sumithra, R.; Paul, S.; , "Using distributed apriori association rule and classical apriori mining algorithms for grid based knowledge discovery," Computing Communication and Networking Technologies (ICCCNT), 2010 International Conference on , vol., no., 29-31 July 2010, pp 1-5

[10] Parag M. Moteria, Dr. Y.R.Ghodasara, "Novel Most Frequent Pattern Mining Approach Using Distributed Computing Environment, International Journal of Engineering Research & Technology (IJERT), Vol. 2, Issue 2, February 2013